

```

PPPPPPPPPPPPP  AAAAAAAA  SSSSSSSSSSSS  RRRRRRRRRRRR  TTTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPPP  AAAAAAAA  SSSSSSSSSSSS  RRRRRRRRRRRR  TTTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPPP  AAAAAAAA  SSSSSSSSSSSS  RRRRRRRRRRRR  TTTTTTTTTTTTTTTT  LLL
PPP    PPP  AAA  AAA  SSS  RRR  RRR  TTT  LLL
PPP    PPP  AAA  AAA  SSS  RRR  RRR  TTT  LLL
PPP    PPP  AAA  AAA  SSS  RRR  RRR  TTT  LLL
PPP    PPP  AAA  AAA  SSS  RRR  RRR  TTT  LLL
PPP    PPP  AAA  AAA  SSS  RRR  RRR  TTT  LLL
PPP    PPP  AAA  AAA  SSS  RRR  RRR  TTT  LLL
PPP    PPP  AAA  AAA  SSS  RRR  RRR  TTT  LLL
PPPPPPPPPPPPP  AAA  AAA  SSSSSSSSS  RRRRRRRRRRRR  TTT  LLL
PPPPPPPPPPPPP  AAA  AAA  SSSSSSSSS  RRRRRRRRRRRR  TTT  LLL
PPPPPPPPPPPPP  AAA  AAA  SSSSSSSSS  RRRRRRRRRRRR  TTT  LLL
PPP  AAAAAAAA  SSS  RRR  RRR  TTT  LLL
PPP  AAAAAAAA  SSS  RRR  RRR  TTT  LLL
PPP  AAAAAAAA  SSS  RRR  RRR  TTT  LLL
PPP  AAA  AAA  SSSSSSSSSSSS  RRR  RRR  TTT  LLL
PPP  AAA  AAA  SSSSSSSSSSSS  RRR  RRR  TTT  LLL
PPP  AAA  AAA  SSSSSSSSSSSS  RRR  RRR  TTT  LLL

```

PPPPPPPP	AAAAAA	SSSSSSS	RRRRRRR	EEEEEEEEE	AAAAAA	VV	VV	AAAAAA	RRRRRRR		
PPPPPPPP	AAAAAA	SSSSSSS	RRRRRRR	EEEEEEEEE	AAAAAA	VV	VV	AAAAAA	RRRRRRR		
PP	PP	AA	AA	SS	RR	RR	EE	AA	AA	RR	RR
PP	PP	AA	AA	SS	RR	RR	EE	AA	AA	RR	RR
PP	PP	AA	AA	SS	RR	RR	EE	AA	AA	RR	RR
PP	PP	AA	AA	SS	RR	RR	EE	AA	AA	RR	RR
PPPPPPPP	AA	AA	SSSSSS	RRRRRRR	EEEEEEEEE	AA	VV	VV	AA	RRRRRRR	
PPPPPPPP	AA	AA	SSSSSS	RRRRRRR	EEEEEEEEE	AA	VV	VV	AA	RRRRRRR	
PP	AAAAAA	SS	RR	RR	EE	AAAAAAA	VV	VV	AAAAAA	RR	RR
PP	AAAAAA	SS	RR	RR	EE	AAAAAAA	VV	VV	AAAAAA	RR	RR
PP	AA	AA	SS	RR	RR	EE	AA	VV	AA	RR	RR
PP	AA	AA	SS	RR	RR	EE	AA	VV	AA	RR	RR
PP	AA	AA	SSSSSSS	RR	RR	EEEEEEEEE	AA	VV	AA	RR	RR
PP	AA	AA	SSSSSSS	RR	RR	EEEEEEEEE	AA	VV	AA	RR	RR

LL	IIIIII	SSSSSSS
LL	IIIIII	SSSSSSS
LL	II	SS
LLLLLLLLL	IIIIII	SSSSSSS
LLLLLLLLL	IIIIII	SSSSSSS

```
1 0001 0 MODULE PASSREAD_VARYING (XTITLE 'Read a varying string'
2 0002 0 IDENT = '1-002' ) = ! File: PASREADVAR.B32 Edit: SBL1002
3 0003 0
4 0004 1 BEGIN
5 0005 1
6 0006 1 ***** ****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 ****
28 0028 1 !
29 0029 1
30 0030 1 ++
31 0031 1 : FACILITY: Pascal Language Support
32 0032 1
33 0033 1 : ABSTRACT:
34 0034 1
35 0035 1 : This module contains procedures which read a varying string
36 0036 1 : from a textfile or a string.
37 0037 1
38 0038 1 : ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 : AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
41 0041 1
42 0042 1 : MODIFIED BY:
43 0043 1
44 0044 1 : 1-001 - Original. SBL 1-April-1981
45 0045 1 : 1-002 - Use PASS$END_READ. SBL 26-May-1982
46 0046 1 --
47 0047 1 --
```

```
49      0048 1 %SBTTL 'Declarations'  
50      0049 1  
51      0050 1   PROLOGUE DEFINITIONS:  
52      0051 1  
53      0052 1  
54      0053 1   REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures  
55      0117 1  
56      0118 1  
57      0119 1   TABLE OF CONTENTS:  
58      0120 1  
59      0121 1  
60      0122 1   FORWARD ROUTINE  
61      0123 1     PASSREAD_VARYING: NOVALUE,  
62      0124 1     PASSREAD_VARYING: NOVALUE;           ! Read from textfile  
63      0125 1           ! Read from string  
64      0126 1  
65      0127 1   MACROS:  
66      0128 1  
67      0129 1     NONE  
68      0130 1  
69      0131 1   EQUATED SYMBOLS:  
70      0132 1  
71      0133 1     NONE  
72      0134 1  
73      0135 1   FIELDS:  
74      0136 1  
75      0137 1     NONE  
76      0138 1  
77      0139 1   OWN STORAGE:  
78      0140 1  
79      0141 1     NONE  
80      0142 1
```

```
82      0143 1 %SBTTL 'PASSREAD_VARYING - Read a varying string from textfile'
83      0144 1 GLOBAL ROUTINE PASSREAD_VARYING (
84      0145 1     STRING: REF VECTOR [, WORD],
85      0146 1     PFV: REF SPASSPFV_FILE_VARIABLE,
86      0147 1     MAX_LENGTH: WORD,
87      0148 1     ERROR
88      0149 1   ) : NOVALUE =
89
90      0150 1
91      0151 1 ++
92      0152 1     FUNCTIONAL DESCRIPTION:
93      0153 1
94      0154 1     This function READs a varying string from the specified textfile.
95      0155 1     Up to MAX_LENGTH characters, or until end-of-line, whichever comes
96      0156 1     first, will be read. The number of characters read is stored in
97      0157 1     the first word of the string.
98      0158 1
99      0159 1     CALLING SEQUENCE:
100     0160 1
101     0161 1     STRING.wvt.v = PASSREAD_VARYING (PFV, mr.r, MAX_LENGTH.rwu.v
102     0162 1           [, ERROR.ja.r])
103     0163 1
104     0164 1     FORMAL PARAMETERS:
105     0165 1
106     0166 1     PFV           - The Pascal File Variable (PFV) passed by reference.
107     0167 1           The structure of the PFV is defined in PASPFV.REQ.
108     0168 1
109     0169 1     MAX_LENGTH    - The maximum length of the string.
110     0170 1
111     0171 1     ERROR          - Optional. If specified, the address to unwind to
112     0172 1           in case of an error.
113     0173 1
114     0174 1     IMPLICIT INPUTS:
115     0175 1
116     0176 1     NONE
117     0177 1
118     0178 1     IMPLICIT OUTPUTS:
119     0179 1
120     0180 1     NONE
121     0181 1
122     0182 1     ROUTINE VALUE:
123     0183 1
124     0184 1     STRING          - The varying string read returned as a function value.
125     0185 1           In accordance with the VAX Procedure Calling Standard,
126     0186 1           the address of the string is passed to this procedure
127     0187 1           as the first parameter, and all other parameters are
128     0188 1           shifted left one position.
129     0189 1
130     0190 1           If an error occurs and is continued by a user handler,
131     0191 1           the value returned is a zero-length string.
132     0192 1
133     0193 1     SIDE EFFECTS:
134     0194 1
135     0195 1           If the f is the standard file INPUT or OUTPUT, it is implicitly opened.
136     0196 1
137     0197 1     SIGNALLED ERROR:
138     0198 1
139     0199 1           NONE
```

```
139      0200 1 !--  
140      0201 1  
141      0202 1  
142      0203 2 BEGIN  
143      0204 2  
144      0205 2 LOCAL  
145      0206 2     CHARS_REMAINING,           ! Number of characters remaining in line  
146      0207 2     FCB: REF $PASSFCB_CONTROL_BLOCK,   ! File Control block  
147      0208 2     PFV_ADDR: VOLATILE,           ! Enable argument  
148      0209 2     UNWIND_ACT: VOLATILE,           ! Enable argument  
149      0210 2     ERROR_ADDR: VOLATILE;          ! Enable argument  
150      0211 2  
151      0212 2  
152      0213 2     ACTUALCOUNT;                  ! Count of arguments  
153      0214 2  
154      0215 2  
155      0216 2     ENABLE  
156      0217 2     PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler  
157      0218 2  
158      0219 2     !+  
159      0220 2     | Get ERROR parameter, if present.  
160      0221 2     !-  
161      0222 2  
162      0223 2     IF ACTUALCOUNT () GEQU 4  
163      0224 2     THEN  
164      0225 2     ERROR_ADDR = .ERROR;           ! Set unwind address  
165      0226 2  
166      0227 2     PFV_ADDR = PFV [PFV$R_PFV];    ! Set PFV address  
167      0228 2  
168      0229 2     !+  
169      0230 2     | Validate PFV and get PFV.  
170      0231 2     !-  
171      0232 2  
172      0233 2  
173      0234 2     !+  
174      0235 2     | Set unwind action to unlock file.  
175      0236 2     !-  
176      0237 2  
177      0238 2     UNWIND_ACT = PASSK_UNWIND_UNLOCK;  
178      0239 2  
179      0240 2  
180      0241 2     !+  
181      0242 2     | Do common initialization.  
182      0243 2     !-  
183      0244 2  
184      0245 2  
185      0246 2  
186      0247 2     !+  
187      0248 2     | Determine number of characters remaining in line.  
188      0249 2  
189      0250 2  
190      0251 2  
191      0252 2  
192      0253 2  
193      0254 2  
194      0255 2  
195      0256 2     !+  
196      0257 2     | CHARS_REMAINING = .FCB [FCBSA_RECORD_END] - .FCB [FCBSA_RECORD_CUR];  
197      0258 2  
198      0259 2  
199      0260 2  
200      0261 2  
201      0262 2  
202      0263 2  
203      0264 2  
204      0265 2  
205      0266 2  
206      0267 2  
207      0268 2  
208      0269 2  
209      0270 2  
210      0271 2  
211      0272 2  
212      0273 2  
213      0274 2  
214      0275 2  
215      0276 2  
216      0277 2  
217      0278 2  
218      0279 2  
219      0280 2  
220      0281 2  
221      0282 2  
222      0283 2  
223      0284 2  
224      0285 2  
225      0286 2  
226      0287 2  
227      0288 2  
228      0289 2  
229      0290 2  
230      0291 2  
231      0292 2  
232      0293 2  
233      0294 2  
234      0295 2  
235      0296 2  
236      0297 2  
237      0298 2  
238      0299 2  
239      0290 2  
240      0291 2  
241      0292 2  
242      0293 2  
243      0294 2  
244      0295 2  
245      0296 2  
246      0297 2  
247      0298 2  
248      0299 2  
249      0290 2  
250      0291 2  
251      0292 2  
252      0293 2  
253      0294 2  
254      0295 2  
255      0296 2  
256      0297 2  
257      0298 2  
258      0299 2  
259      0290 2  
260      0291 2  
261      0292 2  
262      0293 2  
263      0294 2  
264      0295 2  
265      0296 2  
266      0297 2  
267      0298 2  
268      0299 2  
269      0290 2  
270      0291 2  
271      0292 2  
272      0293 2  
273      0294 2  
274      0295 2  
275      0296 2  
276      0297 2  
277      0298 2  
278      0299 2  
279      0290 2  
280      0291 2  
281      0292 2  
282      0293 2  
283      0294 2  
284      0295 2  
285      0296 2  
286      0297 2  
287      0298 2  
288      0299 2  
289      0290 2  
290      0291 2  
291      0292 2  
292      0293 2  
293      0294 2  
294      0295 2  
295      0296 2  
296      0297 2  
297      0298 2  
298      0299 2  
299      0290 2  
300      0291 2  
301      0292 2  
302      0293 2  
303      0294 2  
304      0295 2  
305      0296 2  
306      0297 2  
307      0298 2  
308      0299 2  
309      0290 2  
310      0291 2  
311      0292 2  
312      0293 2  
313      0294 2  
314      0295 2  
315      0296 2  
316      0297 2  
317      0298 2  
318      0299 2  
319      0290 2  
320      0291 2  
321      0292 2  
322      0293 2  
323      0294 2  
324      0295 2  
325      0296 2  
326      0297 2  
327      0298 2  
328      0299 2  
329      0290 2  
330      0291 2  
331      0292 2  
332      0293 2  
333      0294 2  
334      0295 2  
335      0296 2  
336      0297 2  
337      0298 2  
338      0299 2  
339      0290 2  
340      0291 2  
341      0292 2  
342      0293 2  
343      0294 2  
344      0295 2  
345      0296 2  
346      0297 2  
347      0298 2  
348      0299 2  
349      0290 2  
350      0291 2  
351      0292 2  
352      0293 2  
353      0294 2  
354      0295 2  
355      0296 2  
356      0297 2  
357      0298 2  
358      0299 2  
359      0290 2  
360      0291 2  
361      0292 2  
362      0293 2  
363      0294 2  
364      0295 2  
365      0296 2  
366      0297 2  
367      0298 2  
368      0299 2  
369      0290 2  
370      0291 2  
371      0292 2  
372      0293 2  
373      0294 2  
374      0295 2  
375      0296 2  
376      0297 2  
377      0298 2  
378      0299 2  
379      0290 2  
380      0291 2  
381      0292 2  
382      0293 2  
383      0294 2  
384      0295 2  
385      0296 2  
386      0297 2  
387      0298 2  
388      0299 2  
389      0290 2  
390      0291 2  
391      0292 2  
392      0293 2  
393      0294 2  
394      0295 2  
395      0296 2  
396      0297 2  
397      0298 2  
398      0299 2  
399      0290 2  
400      0291 2  
401      0292 2  
402      0293 2  
403      0294 2  
404      0295 2  
405      0296 2  
406      0297 2  
407      0298 2  
408      0299 2  
409      0290 2  
410      0291 2  
411      0292 2  
412      0293 2  
413      0294 2  
414      0295 2  
415      0296 2  
416      0297 2  
417      0298 2  
418      0299 2  
419      0290 2  
420      0291 2  
421      0292 2  
422      0293 2  
423      0294 2  
424      0295 2  
425      0296 2  
426      0297 2  
427      0298 2  
428      0299 2  
429      0290 2  
430      0291 2  
431      0292 2  
432      0293 2  
433      0294 2  
434      0295 2  
435      0296 2  
436      0297 2  
437      0298 2  
438      0299 2  
439      0290 2  
440      0291 2  
441      0292 2  
442      0293 2  
443      0294 2  
444      0295 2  
445      0296 2  
446      0297 2  
447      0298 2  
448      0299 2  
449      0290 2  
450      0291 2  
451      0292 2  
452      0293 2  
453      0294 2  
454      0295 2  
455      0296 2  
456      0297 2  
457      0298 2  
458      0299 2  
459      0290 2  
460      0291 2  
461      0292 2  
462      0293 2  
463      0294 2  
464      0295 2  
465      0296 2  
466      0297 2  
467      0298 2  
468      0299 2  
469      0290 2  
470      0291 2  
471      0292 2  
472      0293 2  
473      0294 2  
474      0295 2  
475      0296 2  
476      0297 2  
477      0298 2  
478      0299 2  
479      0290 2  
480      0291 2  
481      0292 2  
482      0293 2  
483      0294 2  
484      0295 2  
485      0296 2  
486      0297 2  
487      0298 2  
488      0299 2  
489      0290 2  
490      0291 2  
491      0292 2  
492      0293 2  
493      0294 2  
494      0295 2  
495      0296 2  
496      0297 2  
497      0298 2  
498      0299 2  
499      0290 2  
500      0291 2  
501      0292 2  
502      0293 2  
503      0294 2  
504      0295 2  
505      0296 2  
506      0297 2  
507      0298 2  
508      0299 2  
509      0290 2  
510      0291 2  
511      0292 2  
512      0293 2  
513      0294 2  
514      0295 2  
515      0296 2  
516      0297 2  
517      0298 2  
518      0299 2  
519      0290 2  
520      0291 2  
521      0292 2  
522      0293 2  
523      0294 2  
524      0295 2  
525      0296 2  
526      0297 2  
527      0298 2  
528      0299 2  
529      0290 2  
530      0291 2  
531      0292 2  
532      0293 2  
533      0294 2  
534      0295 2  
535      0296 2  
536      0297 2  
537      0298 2  
538      0299 2  
539      0290 2  
540      0291 2  
541      0292 2  
542      0293 2  
543      0294 2  
544      0295 2  
545      0296 2  
546      0297 2  
547      0298 2  
548      0299 2  
549      0290 2  
550      0291 2  
551      0292 2  
552      0293 2  
553      0294 2  
554      0295 2  
555      0296 2  
556      0297 2  
557      0298 2  
558      0299 2  
559      0290 2  
560      0291 2  
561      0292 2  
562      0293 2  
563      0294 2  
564      0295 2  
565      0296 2  
566      0297 2  
567      0298 2  
568      0299 2  
569      0290 2  
570      0291 2  
571      0292 2  
572      0293 2  
573      0294 2  
574      0295 2  
575      0296 2  
576      0297 2  
577      0298 2  
578      0299 2  
579      0290 2  
580      0291 2  
581      0292 2  
582      0293 2  
583      0294 2  
584      0295 2  
585      0296 2  
586      0297 2  
587      0298 2  
588      0299 2  
589      0290 2  
590      0291 2  
591      0292 2  
592      0293 2  
593      0294 2  
594      0295 2  
595      0296 2  
596      0297 2  
597      0298 2  
598      0299 2  
599      0290 2  
600      0291 2  
601      0292 2  
602      0293 2  
603      0294 2  
604      0295 2  
605      0296 2  
606      0297 2  
607      0298 2  
608      0299 2  
609      0290 2  
610      0291 2  
611      0292 2  
612      0293 2  
613      0294 2  
614      0295 2  
615      0296 2  
616      0297 2  
617      0298 2  
618      0299 2  
619      0290 2  
620      0291 2  
621      0292 2  
622      0293 2  
623      0294 2  
624      0295 2  
625      0296 2  
626      0297 2  
627      0298 2  
628      0299 2  
629      0290 2  
630      0291 2  
631      0292 2  
632      0293 2  
633      0294 2  
634      0295 2  
635      0296 2  
636      0297 2  
637      0298 2  
638      0299 2  
639      0290 2  
640      0291 2  
641      0292 2  
642      0293 2  
643      0294 2  
644      0295 2  
645      0296 2  
646      0297 2  
647      0298 2  
648      0299 2  
649      0290 2  
650      0291 2  
651      0292 2  
652      0293 2  
653      0294 2  
654      0295 2  
655      0296 2  
656      0297 2  
657      0298 2  
658      0299 2  
659      0290 2  
660      0291 2  
661      0292 2  
662      0293 2  
663      0294 2  
664      0295 2  
665      0296 2  
666      0297 2  
667      0298 2  
668      0299 2  
669      0290 2  
670      0291 2  
671      0292 2  
672      0293 2  
673      0294 2  
674      0295 2  
675      0296 2  
676      0297 2  
677      0298 2  
678      0299 2  
679      0290 2  
680      0291 2  
681      0292 2  
682      0293 2  
683      0294 2  
684      0295 2  
685      0296 2  
686      0297 2  
687      0298 2  
688      0299 2  
689      0290 2  
690      0291 2  
691      0292 2  
6
```

```

: 196      0257 2      CHAR$_REMAINING = MINU (.CHAR$_REMAINING, .MAX_LENGTH);
: 197      0258 2
: 198      0259 2      !+
: 199      0260 2      | Move characters to user variable and update record pointer.
: 200      0261 2      |-
: 201      0262 2
: 202      0263 2      CH$MOVE (.CHAR$_REMAINING, .FCB [FCBSA_RECORD_CUR], STRING [1]);
: 203      0264 2      FCB [FCBSA_RECORD_CUR] = .FCB [FCBSA_RECORD_CUR] + .CHAR$_REMAINING;
: 204      0265 2
: 205      0266 2      !+
: 206      0267 2      | Set current string length.
: 207      0268 2      |-
: 208      0269 2
: 209      0270 2      STRING [0] = .CHAR$_REMAINING;
: 210      0271 2
: 211      0272 2      !+
: 212      0273 2      | Do end-of-READ processing.
: 213      0274 2      |-
: 214      0275 2
: 215      0276 2      PASS$END_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
: 216      0277 2
: 217      0278 2      RETURN;
: 218      0279 2
: 219      0280 1      END:                                ! End of routine PASSREAD_VARYING

```

```

.TITLE PASSREAD_VARYING Read a varying string
.IDENT \1-002\
```

```

.EXTRN PASSREAD_VARYING
.EXTRN PASSREADV_VARYING
.EXTRN PASS$IO_HANDLER
.EXTRN PASS$VALIDATE_PFV
.EXTRN PASS$INIT_READ, PASS$END_READ
```

```
.PSECT _PASS$CODE,NOWRT, SHR, PIC,2
```

			01FC 00000		.ENTRY PASSREAD_VARYING, Save R2,R3,R4,R5,R6,R7,R8 : 0144
		5E	08 C2 00002		SUBL2 #8, SP : 0203
			7E D4 00005		CLRL ERROR_ADDR
		6D 04 005B	AE 7C 00007		CLRQ UNWIND_ACT
		04	CF DE 0000A		MOVAL 3\$, (FP)
			6C 91 0000F		CMPB (AP), #4
			04 1F 00012		BLSSU 1\$
		08 6E 10	AC D0 00014		MOVL ERROR, ERROR_ADDR : 0224
		AE 08	AC D0 00018	1\$:	MOVL PFV, PFV_ADDR : 0226
		56 08	AC D0 0001D		MOVL PFV, R6 : 0232
		08 00000000G	00 16 00021		JSB PASS\$VALIDATE_PFV
		04 AE	01 D0 00027		MOVL #1, UNWIND_ACT : 0238
		56 08	AC D0 0002B		MOVL PFV, R6 : 0244
		00000000G	00 16 0002F		JSB PASS\$INIT_READ
		04 A7 EC	A7 C3 00035		SUBL3 -20(FCB), -16(FCB), CHAR\$_REMAINING : 0250
		56 F0	50 D0 0003B		MOVL CHAR\$_REMAINING, R0 : 0257
	50	OC AC	10 00 ED 0003E		CMPZV #0, #T6, MAX_LENGTH, R0
			04 1E 00044		BGEQU 2\$
			50 0C		MOVZWL MAX_LENGTH, R0
			56 50		MOVL R0, CHAR\$_REMAINING
			DO 0004A	2\$:	

PASS\$READ_VARYIN Read a varying string
1-002 PASS\$READ_VARYING - Read a varying string from t C 15
16-Sep-1984 02:04:39 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:53 [PASRTL.SRC]PASREAVAR.B32;1 Page 6
(3) PA 1-1002

02	A8	EC	58	04	AC	00	0004D	MOVL	STRING, R8	: 0263
		EC	B7		56	28	00051	MOVC3	CHARS_REMAINING, a-20(FCB), 2(R8)	: 0264
			A7		56	C0	00057	ADDL2	CHARS_REMAINING, -20(FCB)	: 0270
			68		56	B0	0005B	MOVW	CHARS_REMAINING, (R8)	: 0276
			56	08	AC	00	0005E	MOVL	PFV, R6	: 0280
					00	16	00062	JSB	PASS\$END_READ	: 0280
						04	00068	RET		: 0203
							0000 00069	3\$: .WORD	Save nothing	
			50	08	AC	00	0006B	MOVL	8(AP), R0	
			50	04	AC	00	0006F	MOVL	4(R0), R0	
				F4	A0	9F	00073	PUSHAB	ERROR ADDR	
				F8	A0	9F	00076	PUSHAB	UNWIND ACT	
				FC	A0	9F	00079	PUSHAB	PFV_ADDR	
					03	DD	0007C	PUSHL	#3	
					5E	DD	0007E	PUSHL	SP	
			00000000G	7E	04	AC	7D 00080	MOVQ	4(AP), -(SP)	
				00		03	FB 00084	CALLS	#3, PASS\$IO_HANDLER	
						04	0008B	RET		

: Routine Size: 140 bytes, Routine Base: _PASS\$CODE + 0000

: 220 0281 1
: 221 0282 1 !<BLF/PAGE>

```
: 223      0283 1 XSBTTL 'PAS$READV_VARYING- Read varying from string'
: 224      0284 1 GLOBAL ROUTINE PAS$READV_VARYING (
: 225          STRING: REF VECTOR [, BYTE],
: 226          LINE_DSC: REF VECTOR [, BYTÉ],
: 227          MAX_LENGTH: WORD,
: 228          ERROR
: 229      ) : NOVALUE =
: 230
: 231      !++
: 232      !+ FUNCTIONAL DESCRIPTION:
: 233
: 234      This procedure reads a varying-length string from the specified string line.
: 235      Up to MAX_LENGTH characters, or until end-of-line, whichever comes
: 236      first, will be read. The number of characters read is stored in
: 237      the first word of the string.
: 238
: 239      !+ CALLING SEQUENCE:
: 240
: 241      STRING.wvt.r = PAS$READV_VARYING (LINE_DSC.mq.r, MAX_LENGTH.rwu.v
: 242          [, ERROR.ja.r])
: 243
: 244      !+ FORMAL PARAMETERS:
: 245
: 246      0306 1 LINE_DSC      - The string to read from, passed as a class S
: 247          (assumed) descriptor. The length and pointer
: 248          are updated to reflect the unread string.
: 249
: 250      0310 1 MAX_LENGTH    - The maximum length of the string to read.
: 251
: 252      0312 1 ERROR         - Optional. If specified, the address to unwind to
: 253          in case of an error.
: 254
: 255      0315 1 IMPLICIT INPUTS:
: 256
: 257      0317 1 NONE
: 258
: 259      0319 1 IMPLICIT OUTPUTS:
: 260
: 261      0321 1 NONE
: 262
: 263      0323 1 ROUTINE VALUE:
: 264
: 265      0325 1 The value of the string read.
: 266
: 267      0327 1 SIDE EFFECTS:
: 268
: 269      0329 1 NONE
: 270
: 271      0331 1 SIGNALLED ERRORS:
: 272
: 273      0333 1 See PAS$READ_VARYING
: 274
: 275      0335 1 --
: 276
: 277      0337 2 BEGIN
: 278
: 279      0339 2 LOCAL
```

PASS\$READ_VARYIN Read a varying string
1-002 PASS\$READV_VARYING- Read varying from string

E 15
16-Sep-1984 02:04:39 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:53 [PASRTL.SRC]PASREAVAR.B32;1

Page 8
(4)

```
: 280      0340 2      PFV: $PASS$PFV_FILE_VARIABLE,      | Pascal File Variable
: 281      0341 2      RESULT,                      | Result value
: 282      0342 2      ARG_LIST: VECTOR [4, LONG],      | Argument list
: 283      0343 2      PFV_ADDR: VOLATILE,          | Enable argument
: 284      0344 2      UNWIND_ACT: VOLATILE,        | Enable argument
: 285      0345 2      ERROR_ADDR: VOLATILE;        | Enable argument
: 286
: 287      0346 2
: 288      0347 2      BUILTIN ACTUALCOUNT:           | Count of arguments
: 289      0348 2
: 290      0349 2
: 291      0350 2      ENABLE PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);  | Enable error handler
: 292      0351 2
: 293      0352 2
: 294      0353 2
: 295      0354 2      |+ Get ERROR parameter, if present.
: 296      0355 2      |-_
: 297      0356 2
: 298      0357 2      IF ACTUALCOUNT () GEQU 3
: 299      0358 2      THEN
: 300      0359 2      ERROR_ADDR = .ERROR;           | Set unwind address
: 301      0360 2
: 302      0361 2      PFV_ADDR = PFV [PFV$R_PFV];      | Set PFV address
: 303      0362 2
: 304      0363 2
: 305      0364 2      |+ Set up ARG_LIST.
: 306      0365 2      |-_
: 307      0366 2
: 308      0367 2      ARG_LIST [0] = 3;           | Three arguments
: 309      0368 2      ARG_LIST [1] = STRING [0];      | Destination string address
: 310      0369 2      ARG_LIST [2] = PFV [PFV$R_PFV];      | Address of PFV
: 311      0370 2      ARG_LIST [3] = .MAX_LENGTH;        | Maximum string length
: 312      0371 2
: 313      0372 2      |+ Call PASS$DO_READV to do the work, giving it the address of
: 314      0373 2      | PASS$READ_VARYING to call.
: 315      0374 2      |-_
: 316      0375 2
: 317      0376 2
: 318      0377 2      PASS$DO_READV (PFV [PFV$R_PFV], LINE_DSC [0], ARG_LIST, PASS$READ_VARYING);
: 319      0378 2
: 320      0379 2      RETURN;
: 321      0380 2
: 0381 1      END;                                | End of routine PASS$READV_VARYING
```

.EXTRN PASS\$DO_READV

5E		005C 00000	.ENTRY	PASS\$READV_VARYING, Save R2,R3,R4,R6	0284
		28 C2 00002	SUBL2	#40, SP	0337
		7E D4 00005	CLRL	ERROR ADDR	
6D	04	AE 7C 00007	CLRQ	UNWIND ACT	
03	003A	CF DE 0000A	MOVAL	2\$, (FP)	0357
		6C 91 0000F	CMPB	(AP), #3	
		04 1F 00012	BLSSU	1\$	
08	6E	10 AC 00014	MOVL	ERROR, ERROR ADDR	0359
0C	AE	1C AE 9E 00018	MOVAB	PFV, PFV ADDR	0361
		03 DO 0001D	MOVL	#3, ARG_LIST	0367

10 AE 04 AC D0 00021	MOVL STRIN, ARG_LIST+4	0368
14 AE 1C AE 9E 00026	MOVAB PFV, ARG_LIST+8	0369
18 AE 0C AC 3C 0002B	MOVZWL MAX_LENGTH, ARG_LIST+12	0370
54 FF40 CF 9E 00030	MOVAB PASSREAD_VARYING, R4	0377
53 0C AE 9E 00035	MOVAB ARG_LIST, R3	
56 1C AE 9E 00039	MOVAB PFV, R6	
52 08 AC D0 0003D	MOVL LINE_DSC, R2	
00000000G 00 16 00041	JSB PASS\$DO_READV	
04 00047	RET	
0000 00048 2\$:	.WORD Save nothing	0381
50 08 AC D0 0004A	MOVL 8(AP), R0	0337
50 04 A0 D0 0004E	MOVL 4(R0), R0	
D4 A0 9F 00052	PUSHAB ERROR_ADDR	
D8 A0 9F 00055	PUSHAB UNWIND_AC1	
DC A0 9F 00058	PUSHAB PFV_ADDR	
03 DD 0005B	PUSHL #3	
5E DD 0005D	PUSHL SP	
00000000G 7E 04 AC 7D 0005F	MOVQ 4(AP), -(SP)	
00 03 FB 00063	CALLS #3, PASS\$IO_HANDLER	
04 0006A	RET	

: Routine Size: 107 bytes, Routine Base: _FAS\$CODE + 008C

: 322 0382 1
: 323 0383 1 !<BLF/PAGE>

PASSREAD_VARYIN Read a varying string
1-002 PASSREADV_VARYING- Read varying from string
: 325 0384 1 END
: 326 0385 1
: 327 0386 0 ELUDOM

G 15
16-Sep-1984 02:04:39 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:53 [PASRTL.SRC]PASREAVAR.B32;1

Page 10
(5)

: ! End of module PASSREAD_VARYING

PSECT SUMMARY

Name	Bytes	Attributes
_PASS\$CODE	247	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	-----	Symbols	-----	Pages	Processing
	Total	Loaded	Percent	Mapped	Time
-\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	0	0	581	00:01.0
-\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	90	21	33	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:PASREAVAR/OBJ=OBJ\$:\$PASREAVAR MSRC\$:\$PASREAVAR/UPDATE=(ENH\$:\$PASREAVAR)

: Size: 247 code + 0 data bytes
: Run Time: 00:06.6
: Elapsed Time: 00:25.2
: Lines/CPU Min: 3498
: Lexemes/CPU-Min: 11574
: Memory Used: 69 pages
: Compilation Complete

0296 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PASREABOO
LIS

PASREAST1
LIS

PASRESETK
LIS

PASREAREH
LIS

PASREAREG
LIS

PASRESET2
LIS

PASRAB
LIS

PASREADIN
LIS

PASREAREF
LIS

PASREAUAR
LIS

PASREACHA
LIS

PASREADUT
LIS

PASREARENU
LIS

PASREALUNS
LIS

PASREAINT
LIS

PASREASTR
LIS